

Intermediate C# | Patterns, Generics, Threading, Database, Entity Framework, UI & More -

TTCN2155

Explore C# Design Guidelines, Robust Classes, Generics, Lambda Expressions & LINQ, Reflection & Metadata, Resources & Globalization, Basics of Threading and the Basics of Database connectivity.

Duration: 5 Days

Skill Level: Intermediate

Available Format: Instructor-Led Online; Instructor-Led, Onsite In Person ; Blended; On Public Schedule

Developing Applications in C# and .Net Core (Intermediate C# / .Net Core) is a lab-intensive, hands-on course designed to provide you with the skills required to take your C# programming skills to the next level. Throughout the course you'll explore C# Design Guidelines, Robust Classes, Generics, Lambda Expressions & LINQ, Reflection & Metadata, Resources & Globalization, Basics of Threading and the Basics of Database connectivity.

What You'll Learn

Overview

Developing Applications in C# and .Net Core (Intermediate C# / .Net Core) is a lab-intensive, hands-on course designed to provide you with the skills required to take your C# programming skills to the next level. Throughout the course you'll explore C# Design Guidelines, Robust Classes, Generics, Lambda Expressions & LINQ, Reflection & Metadata, Resources & Globalization, Basics of Threading and the Basics of Database connectivity.

Objectives

Our engaging instructors and mentors are highly experienced practitioners who bring years of current "on-the-job" experience into every classroom. Working in a hands-on learning environment, guided by our expert team, attendees will learn to:

- Design and implement classes that adhere to SOLID design principles
- Understand common design patterns
- Implement robust classes with properties, methods, delegates & events
- Understand, use, and implement generics
- Explore generics in the .Net Framework
- Understand and use Language integrated Query (LINQ)
- Understand and use Lambda expressions
- Build dynamic applications using reflection
- Use internationalization to build world friendly applications
- Build multithreaded applications
- Understand the basics of database access
- What's new in .Net Core and C#?

Audience

This is an **intermediate-level** .Net programming course, designed for experienced C# developers

Take Before: Students should have development skills at least equivalent to the following course(s) or should have attended one of these as a pre-requisite:

- **TTCN20483** Programming in C# | Creating Apps in C# and .Net Core (5 days)

Next Steps: We offer a wide variety of follow on courses for .Net development, web development, SOLID design, .Net services, .Net secure coding and others. Please see our **Microsoft Developer Training Suite** list of courses for recommended next steps.

Enhanced Learning Services: Please also ask about our **Pre-Training Class OnRamp & Prep / Primer** offerings, **Skills Gap Assessment Services**, **Case Studies**, **Knowledge Check Quizzes**, **Skills Immersion Programs & Camps**, **Collaborative Mentoring Services** and **Extended Learning Support & Post Training** services.

Pre-Requisites

TTCN20483 Programming in C# / Creating Apps in C# and .Net Core

Agenda

Please note that this list of topics is based on our standard course offering, evolved from typical industry uses and trends. We will work with you to tune this course and level of coverage to target the skills you need most. Course agenda, topics and labs are subject to adjust during live delivery in response to student skill level, interests and participation.

Overview of new Features

- Type specifiers
- Raw Literals
- Top-level statements
- Records
- Global Usings
- Tuples
- Deconstruction

Design Guidelines

- Conventions
- Class Design guidelines
- Exception Design guidelines
- Common Design Patterns

Robust Classes

- Properties
- Functions
- Delegates
- Events

Unit Testing with XUnit

- Utilizing Test Driven Development (TDD)
- Survey of Unit test frameworks
- Using XUnit
- Mocking with Moq

Exceptions

- Overview
- Syntax
- Exceptions vs Return code
- Testing for exceptions

Generics

- Generic Classes
- Generic Functions
- Understanding invariance
- Survey of Generics in the .Net Library

Delegates

- Overview
- Methods as objects
- Generics and Delegates
- Delegates in the library

Events

- Overview
- Defining
- Register/Deregister

Lambda Expressions

- Using lambdas to generate a delegate

Inheritance

- Overview
- Syntax
- Override/Virtual
- Abstract
- Polymorphism

SOLID Principles & Design Patterns

- Overview
- Design Smells
- Single Responsibility
- Open/Close
- Liskov's Substitution Principle
- Using SOLID to correct design smells

Interfaces

- Overview
- Interface motivations and SOLID
- Interface Segregation
- Dependency Inversion
- Default interface methods

LInQ

- Overview
- Extension Methods
- Linq API

- Linq Query

Attributes, Reflection & Metadata

- Reflections basics
- Attributes

Resources & Globalization

- Overview
- Resource Bundles
- Specifying Language/Region

Threads, Tasks, Async/Await

- Creating Threads
- Coordinating threads
- Tasks
- Async/Await

Basics of Entity Framework

- DbContext
- Code first vs Db first
- Entities
- Mapping

Survey of GUI Applications

- MAUI
- Blazor
- Asp.Net

Related Courses

TTCN10975	Getting Started with Programming in C# / .Net Core for Non-Developers
TTCN20486	Developing ASP.NET Core 9.x MVC Web Applications (Intermediate C#)
TTCN20483	Programming in C# / Creating Apps in C# and .Net Core

All course software (limited versions, for course use only), digital courseware files or course notes, labs / data sets and solutions (as applicable) are provided for you in our “easy access / no install required” high-speed remote lab environment. Our tech team works with every student to ensure everyone is set up with working access and ready to go prior to every course start date, ensuring a smooth delivery and great hands-on experience. Please ask for details.

For More Information

Please [contact us](#) or call 844-475-4559 toll free for more information about our training services (instructor-led, self-paced or blended), coaching and mentoring services, public course enrollment or questions, partner programs, courseware licensing options and more.